

## Method for Dynamically Generating Reference Identifiers In Structured Information

### BACKGROUND OF THE INVENTION

#### Technical Field

This invention relates generally to information retrieval in a computer network. More particularly, it relates to an improved method of locating information in a structured information environment.

#### Description Of The Prior Art

It is well known in the computer field to couple a plurality of computer systems into a network of computer systems. By creating a network of computer systems, collective resources available within the network may be shared among users of the network. With the growth of computerized distributed information resources, such as the Internet and private Intranets, sharing of computer resources is now commonly available. Both the Internet and Intranets have become a source for sharing information on medium and larger scale systems and allow users to retrieve vast amounts of electronic information previously unavailable in an electronic medium.

Networked systems utilizing hypertext conventions typically follow a client-server architecture. A client is usually a computer that requests a service provided by another computer known as a server. The server is typically a remote computer system accessible over a communications medium. Based upon requests by the user at the client, the server presents information to the user as responses to the client. The client typically contains a program called a browser that communicates the requests to the server and formats the responses for viewing at the client. The server is typically a remote computer system accessible over a communications medium. The server scans and searches for unprocessed information sources based upon requests by the user. The server presents filtered electronic

information to the user as server responses to the client. The client may be active in a first computer system and the server process may be active in a second computer system. This allows the client and server to communicate with one another over a communications medium thereby allowing multiple clients to take advantage of the information-gathering capabilities of the server. Accordingly, a server is a network computer that executes administrative software that controls access to all or part of the network and its resources, and makes resources available to remote users on the network.

One common use of the Internet and private Intranets is providing access to files within the system. A standard page description language known as the Hypertext Markup Language (HTML) provides basic document formatting and allows the developer to specify links to servers and specific files stored on the servers and their associated media. Retrieval of information is generally achieved through the use of a browser at a client machine. A network path to a server is identified by a Uniform Resource Locator (URL) having a syntax for defining a network connection. When the user of the browser specifies a link via a URL, the client issues a request to a naming service to map a hostname in the URL to a particular network Internet Protocol (IP) address at which the server is located. The naming service returns a list of one or more IP addresses that can respond to the request. Using one of the IP addresses, the browser establishes a connection to a server identified with the IP address. If the server is available, it returns a document or other object formatted according to HTML. Accordingly, browsers have become a primary interface for access to many network and server services.

One problem with retrieving information on the Internet is the amount of time required to sift through the enormous amount of information available to find the files that are of interest for the specific search request. A substantial amount of user time is required to refine search strategies and compile and discard results. Most prior art electronic document delivery systems use HTML formatted documents for search and delivery to the user. In these systems the entire documentation set is often batch processed and contextual information may be incorporated into the documentation directly or by reference. Every time content in the documentation is amended, such as inserted, removed, and/or reorganized, the entire documentation must be reindexed. Accordingly, it is desirable to provide a

method for efficiently generating reference identifiers in electronic documentation that overcomes the drawbacks of the prior art.

## SUMMARY OF THE INVENTION

It is therefore an object of the invention to dynamically generate a reference identifier in an electronic document and to deliver the document with the reference identifier to the user through the hypertext transfer protocol.

A first aspect of the invention is a method for dynamically creating a reference identifier in an electronic document. The document is formatted into a data structure, and the hierarchy of the data structure is followed to reach the root of the data structure. The data structure is traversed from the root until a target object is encountered. A reference identifier is generated based upon a location of the target object in the data structure. The step of traversing the data structure preferably includes incrementing a counter when a specified branch of the data structure is encountered, and clearing the counter when a specified branch of the data structure is closed. The data structure may be recursively traversed from the root. In addition, the reference identifier may be updated to reflect changes in the data structure. The step of updating the reference identifier preferably includes resetting an index for the data structure when content of the data structure is inserted, removed, reorganized, or otherwise amended.

A second aspect of the invention is a computer system having a data structure, a manager responsive to a traverse request of the data structure, and a marker to identify a position of a target object in the data structure. A counter increment responsive to the manager is preferably provided if a specified branch in the data structure matches the traverse request. Alternatively, a counter clearance responsive to the manager may be provided if a specified branch in the data structure is closed. In addition, a modified marker may be provided in response to inserted content, removed content, reorganized content or other amendment to the data structure.

A third aspect of the invention is an article comprising a computer-readable signal bearing medium. The article includes means in the medium for following a hierarchy of a data structure to reach the root of the data structure, means in the medium for traversing the data structure from the root, and means in the medium for identifying a position of a target object in the data structure. The medium is preferably selected from the group consisting of a recordable data storage medium and a modulated carrier signal. The traversal means preferably generates a counter increment responsive to a match of a specified branch in the data structure to a search request. Alternatively, the traversal means may generate a counter clearance responsive to an encounter of a closed branch of the data structure to a search request.

Other features and advantages of this invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a computer system for generating reference identifiers according to the preferred embodiment of this invention, and is suggested for printing on the first page of the issued patent.

FIG. 2 is a flow chart illustrating the process for returning a reference identifier to a client workstation.

FIG. 3 is a flow chart illustrating the first phase for generating content relative identifiers in retrieved data.

FIG. 4 is a flow chart illustrating the second phase for generating content relative identifiers in retrieved data.

FIG. 5 is a flow chart illustrating the third phase for generating content relative identifiers in retrieved data.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

### Technical Background

Recently, it has become common for technical documents to be encoded in a standard markup language, such as Standard Generalized Markup Language (SGML) and Extensible Markup Language (XML). Both the SGML and XML languages utilize clear text character sets such as ASCII or Unicode to store both content and structure of a document. Both of these languages encode the documents into a predefined organizational structure. Therefore, there is no preprocessing of the document required following amendment to the document content. Revisions to the document content and structure are reflected immediately. Context internal to a document is reflected by the document's structure, while context of references to subsets of other documents have no contextual reference, merely a target value. Accordingly, electronic documents encoded in SGML and/or XML format are dynamically updateable and do not require reindexing of contextual information subsequent to amendment of document content.

Organization of electronic documents in a repository is controlled by an XML document that defines how the contents of the library should be organized. The XML document is a data structure in the form of a digital tree. Data structures in the form of trees are efficient tools for supporting searches beginning with a known identifier. A tree is a data structure accessed first at a root node. Each subsequent node can be either an internal node with further subsequent nodes, or an external node with no further nodes existing under the node. An internal node refers to or has links to one or more descending or child nodes and is referred to as the parent of its child nodes, and external nodes are commonly referred to as leaves. The root node is usually depicted at the top of the tree structure and the external nodes are depicted at the bottom of the tree structure. A navigation system interface uses information retrieved from the XML document to construct a data structure tree interface. The lowest level of the tree of the XML document provides a pointer to a document in a document repository. The intermediate levels of the tree are constructed using data retrieved from the electronic document. It is

the intermediate levels of the tree that are dynamically updated through navigation. A reference identifier is a navigational cue that reflects the structure of the hierarchy of a digital tree. When a document is formatted as a digital tree, reference identifiers may be generated to reference a document fragment within a larger document, or a set of documents, based upon the structural organization of the tree. The process of generating reference identifiers in structured information and presenting them to a client workstation requires both navigation and content delivery. Accordingly, navigation of an XML document in combination with the content delivery mechanism provides delivery of reference identifiers directly to a client workstation.

### Technical Details

Fig. 1 is a block diagram 10 of the components of a computer system used in dynamically generating reference identifiers in electronic documentation. There are four components in the system: a client workstation 20, a server 30, a document repository 40, and an SGML language processing tool 50. The server 30 includes a communication module 32 for allowing the server to communicate with the database, a data structure module 34 for enabling traversal of the hierarchy of a data structure, and a viewing module 36 for controlling formatting of electronic documents. The communication module 32 is a document retrieval system for extracting documents or document fragments from the document repository 40, translating extracted data in the documents or document fragments from the document repository 40 from SGML to HTML, and presenting the extracted data to the client workstation 20. Actual delivery of the extracted data to the client workstation 20 is controlled by the server 30. The SGML language process tool 50 translates and formats SGML document content into HTML format for delivery to the client workstation 20 through hypertext transfer protocol (HTTP). The document repository 40 is a database or hierarchy in a file system, such as folders on a hard drive. The database may be a relational database or an object oriented database. Although some of the components of the system may be commercially available, it is the interaction of the commercially available components with the novel modules that allow the system to dynamically generate reference identifiers in electronic documentation.

Fig. 2 is a flow diagram 100 illustrating a sample navigation request and content delivery cycle for returning a reference identifier to a client workstation. The client generates a navigation request 110. The navigation request is received in an HTTP encoded format. The navigation request is received by the communication module of the server 115. Thereafter, the navigation request is transferred from the communication module of the server to the data structure module of the server 120. At step 120, the data structure module converts the navigation request from the HTTP encoded format to string values corresponding to XML elements. All searches in the XML data structure are initiated at the root of the data structure. Therefore, the hierarchy of the XML document is followed until the root of the data structure is attained 123. The data structure module then searches the XML document for matching string values in the relevant attributes of the XML document elements 125. The searches in steps 123 and 125 are preferably conducted recursively. A determination must be made to assess whether the values encountered in the tree traversal are relevant 130 to the predefined elements. If a specified branch of the data structure is encountered, a counter is incremented 133. Each relevant attribute value from each encountered element in the XML document is added to an HTML document with formatting indicating the depth of each element in the XML document tree 135. Retrieved attribute values are built up as HTTP query strings and encoded as a URL target for HTML elements inserted into the HTML document. However, if at step 130 the specified branch of the data structure is determined to be closed, then the counter is cleared 147 and the lowest match of the query is returned to the HTML document 150. The HTML document is returned to the client workstation as new content 145. Following step 135, a determination as to whether traversal of the data structure is complete must be assessed 140. When traversal of the XML document is complete, the HTML document is returned to the client workstation as new content 145. However, if an attribute value does not match an incoming query parameter for its element, that element is not traversed. At the lowest match of the query to the XML document, the data structure enumerates that subtree and returns the data for its children 150, and the HTML document is returned to the client workstation as new content 145. The data in the HTML document returned to the client workstation contains reference identifiers that reflect the structure of the hierarchy of the document.

In addition to generating reference identifiers for retrieved content, content relative reference identifiers in retrieved data can be generated. This process occurs in three phases. Fig. 3 is a flow diagram 200 illustrating the first phase of this process involving processing retrieved content. The first step is for a client to request a particular document or document fragment 210. The request is sent from the client workstation 20 to the server 30. The viewing module 36 of the server converts the client generated query into a database query 215. The viewing module 36 then initiates the retrieval of the document content from the document repository 40 by establishing hierarchical location of the content within the entire document 220. The document content is returned to the viewing module in SGML format 225. Accordingly, Fig. 3 demonstrates the first phase in the process of the content delivery cycle for generating content relative reference identifiers.

Fig. 4 is a flow diagram 250 illustrating the second phase for processing retrieved content. The viewing module 36 monitors the document content returned in step 225 in the first phase for specific attributes to resolve the context of cross references 255. For example, a reference to another document would require inserting correct numbering in a cross-reference hyperlink. A determination must then be made if a cross-reference to a secondary document is present in the document content returned from the viewing module 260. If the determination at step 260 is positive, the hierarchy of the data structure is followed to find the root of the tree 263 and the application program interface accesses the document repository to resolve the context of the cross-reference 265. Once the context for the secondary reference has been established 267, the cross-reference information is inserted into an SGML formatted document and returned to the viewing module 270. The viewing module sends the returned SGML document together with the cross-reference information to the SGML language process tool 275. Accordingly, the second phase for generating content relative identifiers outlines the algorithm for resolving cross-references to secondary documents.

Fig. 5 is a flow diagram 300 illustrating the third phase for processing retrieved content. Following receipt of an SGML document resolving the cross-reference to a secondary document at step 275, the language processing tool translates the SGML document content to HTML format 310,



and the cross-reference information inserted by the viewing modules is converted to a reference identifier(s) according to the SGML element in which they occur or in which the element to which they refer occurs 315. The viewing module delivers the HTML document through the web server to the client workstation 320. In some cases the content of the delivered HTML document may contain a new navigation query. If this occurs, the process returns to step 210 of Fig. 3. Accordingly, the third phase for generating content relative identifiers converts the reference identifiers in the SGML document to HTML format and delivers the HTML document to the client workstation.

The process of generating reference identifiers is the ability to generate a link in a user interface to a document or a section in a document. The reference identifier provides a location of a referenced object in the context of it's parent document. Since SGML and XML formatted documents are hierarchical, amendments to the document are reflected immediately and do not require re-formatting. Amendment to a document may include inserted content, removed content, and reorganized content, as well as other forms of amendment to a document. When a client reloads a browser page, the reference identifiers are updated to reflect changes in the data structure, *i.e.* the XML and SGML documents. Accordingly, since a reference identifier is a link in the user interface to a document or a section in a document, the process of updating a reference identifier to an amended document includes resetting an index for the data structure.

### Advantages Over The Prior Art

The preferred embodiment of the invention provides a method for creating a reference identifier to a target object in a data structure. The method outlined in the preferred embodiment enables dynamic creation of a reference identifier to an electronic document. SGML and XML formatted documents are hierarchical by nature. The format of the documents in either of these languages enables authors to amend the documents without recompiling the documents. Any prior reference identifier to a section within a document is regenerated when a user executes a reload from a client workstation. Accordingly, a reference identifier to a document or a cross-referenced document is dynamic by nature.

## Alternative Embodiments

5